

**Amendments to the Specification**

Please replace the paragraph beginning on page 10, line 5 with the following amended paragraph:

Figure 1 shows the principle of the Distributed Component Object Model (DCOM) security concept as it is implemented by the company Microsoft in its system Windows NT. When a user, who works on a first Computer system 1 (client), wants to access information, for example software or database information, on a second Computer system 2 (server) he has to go through an authentication and authorisation process which is carried out under Windows NT, the operating system for the client 1 and the server 2. In a first step, represented by the arrow 5, the client 1 sends a user-id, domain name and node to the server 2, which responds with a challenge, represented by arrow 6. This challenge is passed, together with the user-id of the applying user, to the Primary Domain Controller (PDC) 3, which in this example is installed in the remote control location 4. This transfer is independently done by the client 1 to PDC 3, shown by arrow 8, and by server 2 to PDC 3, shown by arrow 11. The PDC 3 then calculates the response to the user-id and the challenge by using the hash value of a user individual password, i.e. the value which is received from subjecting a user individual password to a hash function. As explained earlier, a hash function is understood to be a one-way function, that means it may be applied in one direction without any problems, but it is virtually impossible to invert. The response is then transmitted to the client 1 and to the server 2, shown by arrows 9 and 10, respectively. Together with the response a session key is transmitted, which is session individual in order to render it more difficult to figure the system out. Finally, the client 1 transfers the response 7 to the server 2, thereby proving his identity and authorisation.

Please replace the paragraph beginning on page 13, line 15 with the following amended paragraph:

Figure 4 shows an embodiment according to the present invention. A client 50 operating under a first security system 51, e.g. a NT workstation 50 operating under the control of PDC 51. The client 50 sends a request 54-58, consisting at least of a user-id, if necessary also of a domain

name and a node, to the second computer system 64, preferably a server. The access control unit 57, e.g. a DCOM security system on OS/390 as described with Figure 2 above, in the second Computer system 64, running in this example under OS/390, answers with a challenge 55. The first computer system 50 then passes the user-id and the challenge on to the first security system 51 by communication 52. The first security system 51 answers with the return of a first response and, regularly, a session key by communication 53. This first response is calculated by means of a shared secret registered in the first security system 51. The access control unit 57 passes the user-id on to trusted agent 60 by communication 58. The trusted agent 60 then transfers the user-id to the second security system 63, e.g. a RALF for OS/390 via communication 61. This second security system 63 returns the shared secret to the trusted agent 60 via communication 62 which passes it on to the access control unit 57 via communication 59, thereby enabling the latter to calculate a second response and, if required, the session key. The access control unit 57 is able to apply the same rules for calculating a response to a challenge as the first security system 51. Finally, the first computer system 50 passes its first response by communication 56 to the access control unit 57 of the second computer system 64 which compares it with the second response obtained from the trusted agent 60. If the first and second responses are identical the user or client is granted access, otherwise the access is denied.

Please replace the paragraph beginning on page 15, line 18 with the following amended paragraph:

An optional feature of the inventive method, which, however, may weaken the security of the system, is the ability to automatically synchronise passwords across systems and platforms. An exemplary embodiment for this feature is shown in Figure 5. It shows the concept of a password synchronisation between two different security systems. Regularly, the different security systems use different passwords to check the access to their system. If the password of a user for his entry into a first Computer system 70 running under a first security system shall be changed and this change shall be communicated via communication 71 to a second computer system 72 operating under a second, different security system 75 in order to synchronize the passwords under the two security systems, the client 70 sends the user-id, old and new password and the shared secret, e.g. the hash value, derived from the new password, all in encrypted form

to the trusted agent 73. The trusted agent 73 is in communication with the second computer system 72 and may but need not be, as it is shown here, part of the second Computer system 72. The trusted agent 73 decrypts the transmitted encrypted information and passes via communication 78 the new password and the shared secret calculated from the new password to the second security system 75. The result of this process is that the second security system disposes of the full information about the passwords of the first security system for the user concerned. Herewith, the communication between the security systems is rendered much easier, since any confusion resulting from the use of different passwords is avoided.